

WHAT IS CLAIMED IS:

- 1 1. A method for determining whether documents, in a large
2 collection of documents, are near-duplicates, the method
3 comprising:
 - 4 a) for each of at least some of the documents in the
5 large collection of documents, generating at least two
6 fingerprints;
 - 7 b) preprocessing the fingerprints to identify any
8 fingerprints that are associated with only one
9 document; and
 - 10 c) determining whether or not documents are
11 near-duplicate documents based on fingerprints other
12 than those identified as being associated with only
13 one document.
- 1 2. The method of claim 1 wherein the act of determining
2 whether or not documents are near-duplicate documents
3 includes:
 - 4 i) for any two documents, determining whether or
5 not any fingerprints of a first of the two
6 documents matches any fingerprints of a second of
7 the two documents, and
 - 8 ii) if it is determined that a fingerprint of
9 the first of the two documents does match a
10 fingerprint of the second of the two documents,
11 then concluding that the two documents are
12 near-duplicates.
- 1 3. The method of claim 1 wherein the act of generating at
2 least two fingerprints for each of the documents includes:
 - 3 i) extracting parts from the document,

4 ii) hashing each of the extracted parts to
5 generate a hash value for each of the extracted
6 parts,
7 iii) populating a predetermined number of lists
8 with the extracted parts based on their
9 respective hash values, and
10 iv) for each of the predetermined number of
11 lists, determining a fingerprint based on the
12 contents of the list.

1 4. The method of claim 3 wherein the act of hashing each
2 of the extracted parts to generate a hash value for each of
3 the extracted parts uses a hash function that is
4 repeatable, deterministic and not sensitive to state.

1 5. The method of claim 3 wherein the parts extracted from
2 the document are selected from a group of parts consisting
3 of characters, words, sentences, paragraphs and sections.

1 6. The method of claim 3 wherein the parts extracted from
2 the document do not overlap.

1 7. The method of claim 3 wherein the parts extracted from
2 the document overlap.

1 8. The method of claim 3 wherein each of the acts of
2 determining a fingerprint uses a hashing function with a
3 low probability of collision.

1 9. The method of claim 3 wherein the act of determining a
2 fingerprint uses a function that is sensitive to an order
3 of the parts within a list.

1 10. The method of claim 3 wherein the act of determining a
2 fingerprint uses a function that is insensitive to an order
3 of the parts within a list.

1 11. An apparatus for determining whether documents, in a
2 large collection of documents, are near-duplicates, the
3 apparatus comprising:

4 a) a fingerprint generator for generating, for each
5 of the documents in the large collection of documents,
6 at least two fingerprints;
7 b) a preprocessor for identifying any fingerprints
8 that are associated with only one document; and
9 c) a fingerprint comparison facility for determining
10 whether or not documents are near-duplicate documents
11 based on fingerprints other than those identified as
12 being associated with only one document.

1 12. The apparatus of claim 11 wherein the fingerprint
2 generator includes:

3 i) an extractor for extracting parts from the
4 document,
5 ii) a hashing facility for hashing each of the
6 extracted parts to generate a hash value for each
7 of the extracted parts,
8 iii) list population facility for populating a
9 predetermined number of lists with the extracted
10 parts based on their respective hash values, and
11 iv) means for determining a fingerprint for each
12 of the predetermined number of lists, based on
13 the contents of the list.

1 13. A method for clustering documents, the method
2 comprising:
3 a) for each of the documents, generating at least two
4 fingerprints; and
5 b) for each of the documents,
6 i) determining whether or not the document is a
7 near-duplicate of any of previously processed
8 documents, based on fingerprints of the
9 documents,
10 ii) if it is determined that the document is not
11 a near-duplicate of any previously processed
12 document, then associating the document with a
13 unique cluster identifier, and
14 iii) if it is determined that the document is a
15 near-duplicate of a previously processed
16 document, then associating the document with a
17 cluster identifier associated with the previously
18 processed document.

1 14. A method for filtering search results to remove
2 near-duplicates, the method comprising:
3 a) for each of a predetermined number of candidate
4 search results, determining whether the candidate
5 search result is a near-duplicate of another candidate
6 search result; and
7 b) if it is determined that the candidate search
8 result is a near-duplicate of another candidate search
9 result, then rejecting the candidate search result.

1 15. The method of claim 14 wherein the act of determining
2 whether a candidate search result is a near-duplicate of
3 another candidate search result includes

4 i) comparing a cluster identifier of the
5 candidate search result with that of the other
6 candidate search result, and
7 ii) if the cluster identifiers of the two
8 candidate search results match, then concluding
9 that the two candidate search results are
10 near-duplicates.

1 16. The method of claim 15 wherein cluster identifiers of
2 the candidate search results are assigned by:
3 i) determining whether or not a document
4 corresponding to the candidate search result is a
5 near-duplicate of any of previously processed
6 documents,
7 ii) if it is determined that the document
8 corresponding to the candidate search result is
9 not a near-duplicate of any previously processed
10 document, then associating the document with a
11 unique cluster identifier, and
12 iii) if it is determined that the document
13 corresponding to the candidate search result is a
14 near-duplicate of a previously processed
15 document, then associating the document
16 corresponding to the candidate search result with
17 a cluster identifier associated with the
18 previously processed document.

1 17. A method for determining whether two documents are
2 near-duplicates, the method comprising:
3 a) for each of the two documents, generating at least
4 two fingerprints by
5 i) extracting parts from the document,

ii) hashing each of the extracted parts to generate a hash value for each of the extracted parts,

iii) populating at least two lists with the extracted parts based on their respective hash values, and

iv) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list; and

b) determining whether or not the two documents are near-duplicate documents based on their fingerprints.

1 18. The method of claim 17 wherein the act of determining
2 whether or not the two documents are near-duplicate
3 documents includes:

i) determining whether or not any fingerprints of a first of the two documents matches any fingerprints of a second of the two documents, and

ii) if it is determined that a fingerprint of the first of the two documents does match a fingerprint of the second of the two documents, then concluding that the two documents are near-duplicates.

1 19. The method of claim 17 wherein the act of hashing each
2 of the extracted parts to generate a hash value for each of
3 the extracted parts uses a hash function that is
4 repeatable, deterministic and not sensitive to state.

1 20. The method of claim 17 wherein the parts extracted
2 from the document are selected from a group of parts

3 consisting of characters, words, sentences, paragraphs and
4 sections.

1 21. The method of claim 17 wherein the parts extracted
2 from the document do not overlap.

1 22. The method of claim 17 wherein the parts extracted
2 from the document overlap.

1 23. The method of claim 17 wherein the act of determining
2 a fingerprint uses a hashing function with a low
3 probability of collision.

1 24. The method of claim 17 wherein the act of determining
2 a fingerprint uses a function that is sensitive to an order
3 of the parts within a list.

1 25. The method of claim 17 wherein the act of determining
2 a fingerprint uses a function that is insensitive to an
3 order of the parts within a list.

1

1 26. A method, for use in a crawling facility, for reducing
2 processing and bandwidth used, the method comprising:

3 a) for each of the documents, generating at least two

4 fingerprints by

5 i) extracting parts from the document,

6 ii) hashing each of the extracted parts to

7 generate a hash value for each of the extracted
8 parts,

9 iii) populating at least two lists with the

10 extracted parts based on their respective hash

11 values, and

- iv) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list;
- b) determining whether or not the two documents are near-duplicate documents based on their fingerprints;

and

- c) if it is determined that the two documents are near-duplicates, then indicating that one of the two documents is not to be processed during a subsequent crawl.

27. A method for treating broken links to document, the method comprising:

- a) determining whether a link to a first document is broken;
- b) if it is determined that a link to a first document is broken, determining whether there exists a second document that is a near-duplicate of the first document; and
- c) if it is determined that there exists a second document that is a near-duplicate of the first document, then replacing the broken link to the first document with a link to the second document,

wherein the act of determining whether or not there exists a second document is a near-duplicate of the first document is performed by:

i) for each of the documents, generating at least two fingerprints by

- A) extracting parts from the document,
- B) hashing each of the extracted parts to generate a hash value for each of the extracted parts,

22 C) populating at least two lists with the
23 extracted parts based on their respective
24 hash values, and
25 D) for each of the predetermined number of
26 lists, determining a fingerprint based on
27 the contents of the list; and
28 ii) determining whether or not the two documents
29 are near-duplicate documents based on their
30 fingerprints.

1 28. An apparatus for determining whether two documents are
2 near-duplicates, the apparatus comprising:

3 a) a fingerprint generator for generating at least
4 two fingerprints for each of the two documents, the
5 fingerprint generator including
6 i) an extractor for extracting parts from the
7 document,
8 ii) a hashing facility for hashing each of the
9 extracted parts to generate a hash value for each
10 of the extracted parts,
11 iii) a list population facility for populating
12 at least two lists with the extracted parts based
13 on their respective hash values, and
14 iv) means for determining, for each of the
15 predetermined number of lists, a fingerprint
16 based on the contents of the list; and
17 b) a comparison facility for determining whether or
18 not the two documents are near-duplicate documents
19 based on their fingerprints.

1 29. An improved crawling facility, for reducing processing
2 and bandwidth used, the crawling facility comprising:

3 a) a fingerprint generator for generating, for each
4 of the documents, at least two fingerprints, the
5 fingerprint generator including
6 i) an extractor for extracting parts from the
7 document,
8 ii) a hashing facility for hashing each of the
9 extracted parts to generate a hash value for each
10 of the extracted parts,
11 iii) a list population facility for populating
12 at least two lists with the extracted parts based
13 on their respective hash values, and
14 iv) means for determining, for each of the
15 predetermined number of lists, a fingerprint
16 based on the contents of the list;
17 b) a comparison facility for determining whether or
18 not the two documents are near-duplicate documents
19 based on their fingerprints; and
20 c) a document processor, wherein if it is determined
21 that the two documents are near-duplicates, then the
22 document processor indicates that one of the two
23 documents is not to be processed during a subsequent
24 crawl.

1 30. A search filter for processing search results to
2 remove near-duplicates, the search filter comprising:
3 a) a near-duplicate determination facility for
4 determining, for each of a predetermined number of
5 candidate search results, whether the candidate search
6 result is a near-duplicate of another candidate search
7 result; and
8 b) a filter for rejecting the candidate search result
9 if it is determined that the candidate search result

10 is a near-duplicate of another candidate search
11 result.

1 31. The search filter of claim 30 wherein the
2 near-duplicate determination facility includes a comparison
3 facility for comparing a cluster identifier of the
4 candidate search result with that of another candidate
5 search result, and wherein if the cluster identifiers of
6 the two candidate search results match, then it is
7 concluded that the two candidate search results are
8 near-duplicates.

1 32. A machine-readable medium having stored thereon a
2 plurality of records, each of the records comprising:
3 a) a first field for storing a document identifier;
4 and
5 b) a plurality of lists, each of the plurality of
6 lists containing elements of a document identified by
7 the document identifier stored in the first field,
8 wherein a hash function is used to determine
9 which of the plurality of lists each of the elements will
10 be contained in.

1 33. A machine-readable medium having stored thereon a
2 plurality of records, each of the records comprising:
3 a) a first field for storing a document identifier;
4 and
5 b) a plurality of fingerprints, wherein each of the
6 fingerprints is a low collision probability hash
7 function of elements contained in a corresponding
8 list, and wherein the elements are elements of a

9 document identified by the document identifier stored
10 in the first field.

1 34. A machine-readable medium having stored thereon
2 machine-executable instructions which, when executed by a
3 machine:

4 a) extract parts from a document,
5 ii) hash each of the extracted parts to generate a
6 hash value for each of the extracted parts,
7 iii) populate a predetermined number of lists with
8 the extracted parts based on their respective hash
9 values, and
10 iv) for each of the predetermined number of lists,
11 determine a fingerprint based on the contents of the
12 list.

1 35. A method for generating at least two fingerprints for
2 a document comprising:

3 a) extracting parts from the document;
4 b) hashing each of the extracted parts to generate a
5 hash value for each of the extracted parts;
6 c) populating a predetermined number of lists with
7 the extracted parts based on their respective hash
8 values; and
9 d) for each of the predetermined number of lists,
10 determining a fingerprint based on the contents of the
11 list.

1 36. The method of claim 35 wherein each of the lists has
2 an associated hashing function,

3 wherein each of the extracted parts can be contained
4 in none of the lists, one of the lists, or more of the
5 lists based on the hash functions for the lists.

1 37. The method of claim 36 wherein for each hash function
2 is dynamically adjusted such that the probability that the
3 hash function will populate its associated list with a part
4 decreases as the size of the document increases.

1 38. A method comprising:

2 a) determining whether there exists a second document
3 that is a near-duplicate of a first document; and
4 b) indexing the first document but not the second
5 document,

6 wherein the act of determining whether or not
7 there exists a second document is a near-duplicate of the
8 first document is performed by:

9 i) for each of the documents, generating at
10 least two fingerprints by
11 A) extracting parts from the document,
12 B) hashing each of the extracted parts to
13 generate a hash value for each of the
14 extracted parts,
15 C) populating at least two lists with the
16 extracted parts based on their respective
17 hash values, and
18 D) for each of the predetermined number of
19 lists, determining a fingerprint based on
20 the contents of the list; and
21 ii) determining whether or not the two documents
22 are near-duplicate documents based on their
23 fingerprints.

1 39. A method for determining whether two documents are
2 near-duplicates, the method comprising:
3 a) for each of the two documents, generating at least
4 two fingerprints; and
5 b) determining whether or not the two documents are
6 near-duplicate documents by
7 i) determining whether or not any one of the
8 fingerprints of a first of the two documents
9 matches any one of the fingerprints of a second
10 of the two documents, and
11 ii) if it is determined that any one fingerprint
12 of the first of the two documents does match any
13 one fingerprint of the second of the two
14 documents, then concluding that the two documents
15 are near-duplicates.

1 40. A method for determining whether two objects are
2 near-duplicates, the method comprising:
3 a) for each of the two objects, generating at least
4 two fingerprints by
5 i) extracting features from the object,
6 ii) hashing each of the extracted features to
7 generate a hash value for each of the extracted
8 features,
9 iii) populating at least two lists with the
10 extracted features based on their respective hash
11 values, and
12 iv) for each of the predetermined number of
13 lists, determining a fingerprint based on the
14 contents of the list; and

15 b) determining whether or not the two objects are
16 near-duplicates based on their fingerprints.

1 41. The method of claim 40 wherein each of the two objects
2 is a word, and
3 wherein the extracted features define context vectors.

1 42. The method of claim 40 wherein each of the two objects
2 is a word, and
3 wherein, in each case, the extracted features are
4 words that frequently occur in close proximity to the word.

1 43. The method of claim 40 wherein the two objects are
2 words, and
3 wherein if the two objects are determined to be near
4 duplicates, then determining the two words to be synonyms.

1 44. A method for determining whether a first document and
2 a second document in a collection of documents are
3 near-duplicates, the method comprising:
4 a) for each of the documents in the collection of
5 documents, generating at least two fingerprints; and
6 b) concluding that the first and second documents are
7 near-duplicates if any one of the at least two
8 fingerprints of the first document matches any one of
9 the at least two fingerprints of the second document,
10 wherein documents in the collection of documents
11 without any common fingerprints are not checked to
12 determine whether or not they are near duplicates.

1 45. The method of claim 44 further comprising:

- 2 a2) for each of the documents in the collection of
- 3 documents, generating a document-fingerprint pair for
- 4 each of the at least two fingerprints; and
- 5 a3) sorting the fingerprint-document pairs based on
- 6 values of the fingerprints.